

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-186673

(43)Date of publication of application : 04.07.2003

(51)Int.CI.

G06F 9/44

(21)Application number : 2002-253270

(71)Applicant : HITACHI LTD

(22)Date of filing : 30.08.2002

(72)Inventor : ODAKA TOSHIYUKI  
ARINDAM SAHA

(30)Priority

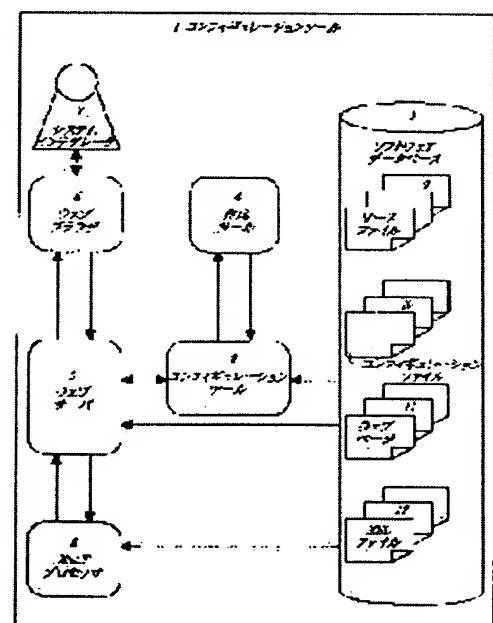
Priority number : 2001 028045 Priority date : 20.12.2001 Priority country : US

## (54) INTEGRATION METHOD OF COMPUTER SYSTEM COMPONENTS, AND CONFIGURATION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To bridge the gap between a system integrator and completed software.

SOLUTION: The integration of application programs including middleware generated by the system integrator 7 from individual components, capable of being parameterized originates from a creator and an environment and is reused. Software creation tools 4 (compiler, assembler, linker and the like) are used, and a tag describes a parameter for software configuration. A specific configuration tool 2 automatically reads out a file 10, presents the choices for the integrator 7, and based on the choice, creates an integrated program. The configuration in a local environment and a configuration are in a distributed environment via the use of a server 5 and a browser 6. The browser 6 having a function tied to the integrator 7 and the distributed environment is independent of the specific component.



## LEGAL STATUS

[Date of request for examination] 24.08.2005

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-186673

(P2003-186673A)

(43)公開日 平成15年7月4日(2003.7.4)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 9/44

識別記号

F I  
G 0 6 F 9/06

テマコード(参考)

6 2 0 A 5 B 0 7 6  
6 2 0 B  
6 2 0 H  
6 2 0 K

審査請求 未請求 請求項の数27 OL (全 17 頁)

(21)出願番号 特願2002-253270(P2002-253270)  
(22)出願日 平成14年8月30日(2002.8.30)  
(31)優先権主張番号 10/028, 045  
(32)優先日 平成13年12月20日(2001.12.20)  
(33)優先権主張国 米国(US)

(71)出願人 000005108  
株式会社日立製作所  
東京都千代田区神田駿河台四丁目6番地  
(72)発明者 小高俊之  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 アリンダム・サハ  
アメリカ合衆国、カリフォルニア州  
94086-2440、サニーヴェール、アシロマ  
ー・テラス 970-4  
(74)代理人 100080001  
弁理士 筒井 大和

最終頁に続く

(54)【発明の名称】コンピュータシステム構成要素の統合方法、およびコンフィギュレーションシステム

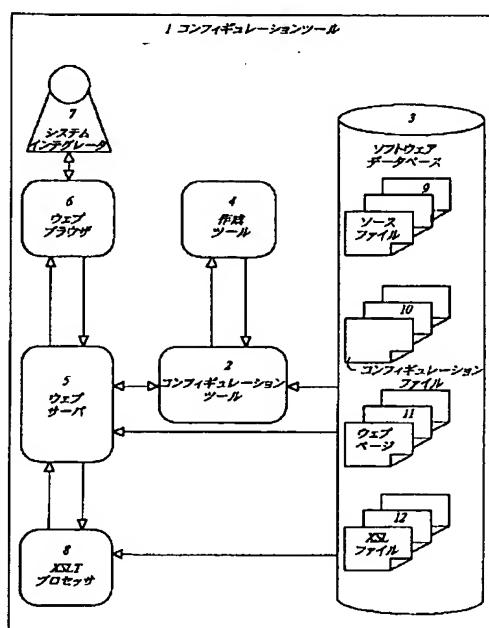
(57)【要約】

【課題】システムインテグレータと完成ソフトウェアとのギャップを埋める。

【解決手段】パラメータ化可能な個々の構成要素からシステムインテグレータ7で作成されたミドルウェアを含め、アプリケーションプログラムの統合は創作者と環境に由来し再利用される。ソフトウェア作成ツール4

(コンパイラ、アセンブラー、リンカー等)が使われ、タグはソフトウェアコンフィギュレーション用のパラメータを記す。特定のコンフィギュレーションツール2はファイル10を自動的に読み出し、インテグレータ7の選択肢を示し、選択肢に基づき統合プログラムを作成する。局所的環境内のコンフィギュレーションに加えコンフィギュレーションはサーバ5とブラウザ6の使用を介して分散環境内にある。インテグレータ7と分散環境に連動する機能を持つブラウザ6は特定の構成要素から独立である。

図 1



## 【特許請求の範囲】

【請求項1】 システムインテグレータと連動するマシンによって実行され、複数の完成された実行可能なプログラムを構成要素として統合して、統合されたプログラムを作成する方法であって、

構成要素に関する複数の選択肢、および、前記複数の完成された実行可能なプログラムの統合用に利用されるパラメータに関する複数の選択肢を選択および生成するためのスクリプトを呼び出す工程と、

前記構成要素の選択肢とパラメータの選択肢とを前記システムインテグレータに提供する工程と、

前記システムインテグレータから構成要素およびそのパラメータの選択肢を受信する工程と、

その後に、前記受信された選択肢に依存して前記完成された実行可能なプログラムから前記統合されたプログラムを作成する工程とを有することを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項2】 請求項1に記載の方法において、前記選択肢に基づいてコンフィギュレーション情報を前記システムインテグレータに提供する工程と、表示された前記コンフィギュレーション情報の確認を前記システムインテグレータからリクエストする工程と、システムインテグレータ確認の受信に応答して、作成する前記工程を実行する工程とを更に有することを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項3】 請求項2に記載の方法において、提供する前記工程のそれぞれが、前記システムインテグレータに対して前記選択肢のそれぞれを表示することを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項4】 請求項3に記載の方法において、提供する前記工程、リクエストする前記工程、および、受信する前記工程のそれぞれが、マークアップ言語を備えたウェブブラウザを介して前記システムインテグレータと連動することを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項5】 請求項3に記載の方法において、選択する前記工程のそれぞれが、ネットワーク上のウェブブラウザを介して、分散環境内の記憶装置への双方向通信を含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項6】 請求項5に記載の方法において、選択する前記工程のそれぞれが、前記分散環境内の前記記憶装置により得られたコンフィギュレーションファイルから、マークアップ言語での前記選択肢のうち対応するものを生成することを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

## 【請求項7】 請求項1に記載の方法において、

選択する前記工程が、ネットワーク上のウェブブラウザを介して、分散環境内の記憶装置への双方向通信を含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項8】 請求項7に記載の方法において、生成する前記工程が、前記分散環境内の前記記憶装置により得られたコンフィギュレーションファイルから、マークアップ言語での前記選択肢のうち対応するものを生成することを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項9】 請求項7に記載の方法において、前記統合されたプログラムによって実行される統合のタイプに関する複数の選択肢を選択する工程と、前記実行される統合のタイプに関する複数の選択肢を前記システムインテグレータに提供する工程と、前記実行される統合のタイプに関する選択肢の中から、前記システムインテグレータからの選択肢を受信する工程と、

統合のタイプに関して受信された選択肢に依存して、前記完成された実行可能なプログラムをダウンロードする工程とを更に有することを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項10】 請求項1に記載の方法において、前記システムインテグレータによって行われた選択と、前記システムインテグレータによって行われなかった選択との表示を前記システムインテグレータに提供し、確認をリクエストする工程と、その後に、システムインテグレータ確認の受信に応答して、前記統合されたプログラムを作成する前記工程を実行する工程とを更に有することを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項11】 請求項1に記載の方法において、選択する前記工程のそれぞれが、コンフィギュレーションファイルを読み出すスクリプトを記憶装置から検索し、前記スクリプトを呼び出して前記生成を実行し、作成する前記工程が、記憶装置からスクリプトと作成ツールとを検索し、呼び出すことを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項12】 請求項1に記載の方法において、前記システムインテグレータに複数の選択肢を提供する前記工程が、前記複数の完成された実行可能なプログラムが格納されるマイクロプロセッサコアの選択肢と、コンパイラの選択肢と、アセンブラーの選択肢と、リアルタイムオペレーティングシステムの選択肢と、速度の選択肢と、性能、電力消費、およびコードサイズに関するコンフィギュレーション情報を含めて前記統合されたプログラムにおける構成要素用のパラメータの選択肢とを提

供することを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項13】 請求項1に記載の方法において、選択する前記工程が、コンフィギュレーションファイルを読み出すスクリプトを記憶装置から検索すること、および、少なくとも、提供する前記工程によって使用するためのコンフィギュレーションファイルを記述するマークアップ言語コードを記憶装置から検索するためのスクリプトを実行することを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項14】 請求項13に記載の方法において、選択する前記工程が、分散環境内に前記コンフィギュレーションファイルを格納することを含むことを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項15】 請求項1に記載の方法において、前記システムインテグレータに複数の選択肢を提供する前記工程と、受信する前記工程とがそれぞれ、媒体のタイプ、プロセッサの識別、最適化のレベル、および、エンディアンの選択肢用に行われるることを特徴とするシステムインテグレータと連動するマシンによって実行される方法。

【請求項16】 コンピュータシステムで使用し、プログラムを統合する際にシステムインテグレータとインターフェースするコンフィギュレーションツールであって、請求項1に記載の前記方法を実行するためのコードの物理的手段を有する記憶媒体を備えたことを特徴とするコンフィギュレーションツール。

【請求項17】 コンピュータシステムで使用し、プログラムを統合する際にシステムインテグレータとインターフェースするコンフィギュレーションツールであって、請求項3に記載の前記方法を実行するためのコードの物理的手段を有する記憶媒体を備えたことを特徴とするコンフィギュレーションツール。

【請求項18】 コンピュータシステムで使用し、統合されたプログラムを作成する際にシステムインテグレータとインターフェースするコンフィギュレーションツールであって、請求項5に記載の前記方法を実行するためのコードの物理的手段を有する記憶媒体を備えたことを特徴とするコンフィギュレーションツール。

【請求項19】 コンピュータシステムで使用し、プログラムを統合する際にシステムインテグレータとインターフェースするコンフィギュレーションツールであって、請求項6に記載の前記方法を実行するためのコードの物理的手段を有する記憶媒体を備えたことを特徴とするコンフィギュレーションツール。

【請求項20】 分散環境内で作動するために、請求項16に記載の前記コンフィギュレーションツールを備え

たコンフィギュレーションシステムにおいて、前記コンフィギュレーションツールに結合される少なくとも1つのコンピュータと、前記システムインテグレータと、選択および受信する前記工程用の前記記憶媒体とにインタフェースするための前記コンフィギュレーションツール、および、前記コンピュータに結合されたウェブブラウザと、提供する前記工程用の前記システムインテグレータとインターフェースするための前記ウェブブラウザに結合されたディスプレイと、前記コンフィギュレーションツールに結合されたソフトウェア作成ツールとを更に備え、前記記憶媒体が、分散環境内に存在し、マシン読み出し可能形式で構成要素仕様のコンフィギュレーションファイルの物理的な実装を有し、前記複数の完成された実行可能なプログラム用のマシン読み出し可能な構成要素ファイルを含み、前記記憶媒体が、提供および受信する前記工程の間に、前記システムインテグレータとインターフェースするためのマシン読み出し可能なディスプレイページの物理形式を有し、前記コンフィギュレーションツールに結合されたソフトウェアビルドツールを更に有することを特徴とするコンフィギュレーションシステム。

【請求項21】 請求項20に記載のコンフィギュレーションシステムにおいて、

前記コンフィギュレーションファイルが、媒体のタイプと、実行可能なプログラムが格納されるプロセッサコアと、コンパイラおよびアセンブラーのオプションと、リアルタイムオペレーティングシステムと、速度最適化のレベルと、前記複数の完成された実行可能なプログラムのパラメータとから成り、

前記記憶媒体が、媒体のタイプと、プロセッサコアと、速度最適化のレベルと、前記複数の完成された実行可能なプログラムのパラメータとの選択肢用に、マークアップ言語で前記ページ形式の物理的実装を格納することを特徴とするコンフィギュレーションシステム。

【請求項22】 複数の完成された実行可能なプログラムからプログラム統合する際にシステムインテグレータとインターフェースし、コンフィギュレーションファイルを有するコンピュータシステムで使用するコンフィギュレーションツールであって、

システムインテグレータのコマンドの入力に応答して、コンフィギュレーションファイルを読み出す手段と、前記システムインテグレータへのコンフィギュレーション選択肢として、読み出されたコンフィギュレーションファイルを記述する手段と、

前記統合されたプログラム用のコンフィギュレーションデータをフォーマットし、それを前記システムインテグレータに送り、前記コンフィギュレーションデータの確認をリクエストする手段と、

前記複数の完成された実行可能なプログラムを記憶装置からダウンロードすること、および、システムインテグレータの選択肢に対応することを制御する手段と、システムインテグレータの確認の受信に応答して、前記構成要素から前記統合されたプログラムを作成する手段とを有することを特徴とするコンフィギュレーションツール。

【請求項23】 分散環境内で作動するために、請求項22に記載の前記コンフィギュレーションツールを備えたコンフィギュレーションシステムであって、前記コンフィギュレーションツールに結合される少なくとも1つのコンピュータと、

前記システムインテグレータとインターフェースするためのディスプレイベージ形式の物理的実装を有する記憶媒体と、

前記コンフィギュレーションツールに結合されたソフトウェア作成ツールと、

分散環境内の記憶媒体と前記システムインテグレータとにインターフェースするための前記コンフィギュレーションツール、および、前記コンピュータに結合されたウェブブラウザとを更に備えることを特徴とするコンフィギュレーションシステム。

【請求項24】 請求項23に記載のコンフィギュレーションシステムにおいて、

マークアップ言語で少なくとも幾つかの前記コンフィギュレーションファイルの物理的実装を有する記憶媒体を更に備えることを特徴とするコンフィギュレーションシステム。

【請求項25】 請求項24に記載のコンフィギュレーションシステムにおいて、

前記システムインテグレータからの選択肢の入力に応答して、コンフィギュレーションファイルを読み出す方法を物理的に実施するスクリプトの物理的な実装を有する記憶媒体を更に備えることを特徴とするコンフィギュレーションシステム。

【請求項26】 請求項23に記載のコンフィギュレーションシステムにおいて、

前記システムインテグレータからの選択肢の入力に応答して、コンフィギュレーションファイルを読み出す方法を物理的に実施するスクリプトの物理的な実装を有する記憶媒体を更に備えることを特徴とするコンフィギュレーションシステム。

【請求項27】 請求項22に記載のコンフィギュレーションシステムにおいて、

ダウンロードを制御する前記手段が、分散環境内の前記記憶装置と通信することを特徴とするコンフィギュレーションシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、システムインテグ

レータによって、2つ以上の以前に開発されたアプリケーションプログラム、すなわち、完成された実行可能なアプリケーションプログラムの統合に関する。

【0002】

【従来の技術】 プログラムを作成するソフトウェア開発者を手助けする装置は存在するが、そのような装置はシステムインテグレータのレベル用に設計されていない。現在、システムインテグレータは、單一マシン上でプログラムを統合するために働く。しかしながら、一般的に、コンピュータシステムは様々に分散されている。システムインテグレータはミドルウェアを作成して、複数の完成されて実行可能アプリケーションプログラムを統合する。

【0003】 一般に、システムインテグレータの観点から、コンピュータシステムは、次のものから成っていると言える：

- I. アプリケーションプログラム
- II. ミドルウェア
- III. オペレーシングシステム
- IV. ハードウェア。

【0004】 通常、ミドルウェアは、2つの構成要素アプリケーションプログラムの間に、または、アプリケーションプログラムとオペレーティングシステムとの間に置いてある。または、将来に、オペレーティングシステムの一部になるかもしれない。ミドルウェアは、アプリケーションプログラムの間で、および／またはアプリケーションプログラムとオペレーティングシステムとの間で、情報を翻訳する。ミドルウェアを含んだコンピュータシステムの一部が、アセンブラー、コンパイラ、または言語翻訳プログラムをまた含むこともある。このコンピュータシステムはビデオ会議用であるかもしれない。

【0005】 従って、必要とされるハードウェアに関して、ミドルウェアは、ソフトウェアのうち2つ以上の構成要素を統合する。こうして、ミドルウェアは、機能をアプリケーションプログラムに提供し、そのアプリケーションプログラムを、基礎をなすハードウェアに結合する。ミドルウェアはアプリケーションレベルに関数を呼び出し、応答としてミドルウェアへの、および、ミドルウェアを通してのプログラムフローが存在する。統合する際に、cpuの選択によりソフトウェア構成要素のために利用するアセンブリ言語、特定の機械語、または特定の命令セットが決定される。アプリケーションプログラムの選択は、選択されたアセンブリ言語の利用を含んでいる。オペレーティングシステムレベルからのリアルタイム特性がミドルウェアに提供されるので、分離していることに加えてミドルウェアは一種のオペレーティングシステムに、または、そのオペレーティングシステムの一部になることもある。そのようなリアルタイム特性は、ミドルウェアのライブラリに対するシステムコールを含んでいる。

【0006】ハードウェアレベルはミドルウェアに、メモリサイズ、機械命令、および機械アーキテクチャ、たとえばキャッシュのレベル、電力消費性能に関する情報、速度性能に関する情報、コードサイズ、スタティック性能に関する情報、および、ダイナミック性能に関する情報を提供する。

【0007】既存のソフトウェア作成ツールは、ソフトウェア開発者および設計者によって利用され、彼らはXML（拡張マークアップ言語の頭文字）を使って、情報を構成および提供する際に多くのフレキシビリティを提供するカストマイズタグを作り出すことができる。

#### 【0008】

【発明が解決しようとする課題】本発明は、特別のコンフィギュレーションツールを使って、既存の作成ツールとソフトウェアデータベースとにインタフェースし、使用的するシステムインテグレータを手助けし、また、ウェブブラウザおよびマークアップ言語を使って分散環境までシステムコンフィギュレーションを広げる。

【0009】本発明者は、プログラムの統合に関する問題を分析し、その問題の原因を識別および分析し、その問題に対して解決法を提供した。プログラムの統合は、たとえば、ミドルウェア、または、オペレーティングシステムの一部分の作成を含むこともある。問題の分析、原因の識別および分析、および、解決法の提供はそれぞれ本発明の一部であり、以下説明される。

【0010】公知の「マイクファイル」を介してのソフトウェアコンフィギュレーションは多くの制限を伴う。ユーザは、コンフィギュレーションにとって必要な一部だけでなくファイル全体のコンテンツを見ることができるので、マイクファイルの取扱いは時々複雑になる。マイクロソフトのビジュアルC++のような大抵のIDE（統合開発環境）ツールは例外でない。マイクファイルおよびIDEツールは、おそらく、ソフトウェア開発者にとって有用であるが、システムインテグレータにとって役立たない。本発明に基づくシステムは、システムインテグレータと、完成ソフトウェアとの間のギャップを埋める。

【0011】ミドルウェアは、できるだけ早く特定されるべき多くの要件を有している。たとえば、ミドルウェアが格納されるマイクロプロセッサコア、コンパイラオプション、リアルタイムのオペレーティングシステム（RTOS）、速度（応答時間）、および、それがソフトウェアだけの解決法、ハードウェアだけの解決法、または、両者を利用した解決法であるか否か；これら全部が、システム要件に依存するシステムインテグレータによって選択されることもある。

【0012】ミドルウェアの構成要素は、通常、パラメータ化されている。それ故に、ソフトウェアコンフィギュレーションは、秩序だった方法で、これらのパラメータを受け入れなければならない。そのパラメータまたは

コンフィギュレーションに関する構成要素および情報は、システムインテグレータが統合のために利用する単一マシン上で必ず利用できるわけではない；その代わりに、これらの構成要素は、多重のネットワークに接続されたマシンシステム上で分散されることもある。

【0013】ミドルウェアの要件は、単一のコンピュータチップ上で多重機能を統合することを伴った現代のシステムオンチップ（SoCs）の要件と幾つかの類似点を有している。SoCsと全く同様に、個々のミドルウェア構成要素は、本質的に異なる創作者および環境に由来し、理想的には何度も何度も再利用されるべきである。

【0014】現在、ある場合には、プログラマーは、プログラマのような処理方法、たとえば#if, #ifdef, #else, #elif, および#endifを使ってコードを分割する。それに対して、他の場合には、プログラマーは、全ての条件用に分離されたファイルを作り出す。これにより、システムインテグレータ用のもつと複雑なコンフィギュレーションの処理がなされ、ミドルウェアの柔軟性がなくなると共に規模変更がしにくくなり、それがスタンドアローンマシンに限定されてしまう。プログラマは、Cコンパイラのプリプロセッサに対してのコマンドであり、条件コンパイルコマンドであると言われることがあった。

【0015】本発明は、システムインテグレータにより作成されてアプリケーションソフトウェアの構成要素を統合し、それによって、あるターゲットハードウェアに対してアプリケーションソフトウェアの構成要素のうち幾つかまたは全部を最適化するミドルウェアに関する。

【0016】統合の処理の際には、ソフトウェアコンフィギュレーションを簡単にすることが必要とされる。統合における最近の技術は、複雑なだけでなく、システムインテグレータに適切な注意を払わずにソフトウェア開発者だけで自分達だけのために考え出された。ソフトウェアのコンフィギュレーションシステムを最大限に利用するのは、ソフトウェア開発者ではなくてシステムインテグレータである。本発明の問題解決部分は、確認された原因に取り組んで分析することにより、確認された問題の緩和に着手している。

【0017】更に、インターネットの急増により、ソフトウェアの統合がもはやスタンドアローンのマシンに制限されない。ますます、統合される構成要素が、ローカルエリアネットワーク（LAN）とワイドエリアネットワーク（WAN）との両方によって接続された分散環境に存在する。最近のソフトウェアコンフィギュレーションは、分散構成要素を取り扱う能力を欠いている。本発明の一態様は、このような問題を緩和する。

#### 【0018】

【課題を解決するための手段】一態様は、特定のコンフィギュレーションツールおよびソフトウェアデータベー

スを有するソフトウェアのコンフィギュレーションシステムを使用する。その態様は、既存のソフトウェア作成ツール（たとえばコンパイラ、アセンブラー、リンカー、翻訳ソフトなど）、XMLのような既存のインタフェース言語、既存のウェブサーバ、および、既存のウェブブラウザを利用する。既存のソフトウェア作成ツール、既存のインタフェース言語、既存のウェブサーバ、および、既存のウェブブラウザは全て、利用可能であり、公知なものである。これらの構成要素は分散され、たとえば、コンピュータネットワーク環境によって接続されていることもあり、または、局所的に配置されていることもある。

【0019】ソフトウェアデータベース内に格納されたコンフィギュレーションファイル（ハードウェアまたはソフトウェア用の機械読み出し可能な動作仕様を含んだファイル、または、構成要素のパラメータに関する情報を含んだファイル）は、態様に係る既存の言語内に書き込まれる。既存の言語としてXML（拡張マークアップ言語）が好適である。この態様は、本明細書の中でソフトウェア・コンフィギュレーション・マークアップ言語（SCML）と呼ばれるソフトウェアコンフィギュレーション用のパラメータを記載するために使用される一組のタグをXMLで含み、そのソフトウェアコンフィギュレーションは、二種類のコンフィギュレーションファイル：1）それぞれ個々のモジュールすなわち構成要素用のコンフィギュレーションファイルと、2）マイクファイル、すなわちソフトウェアコンフィギュレーションの既存の方法と同様に、全プログラム（またはミドルウェア）用のコンフィギュレーションファイルから成る。

【0020】一態様に含まれる特定なコンフィギュレーションツールは、SCMLタグセット内にコンフィギュレーションファイルを自動的に読み出すスクリプトを使い、システムインテグレータに幾つかの選択肢を示し、システムインテグレータの選択肢に基づいて他の選択肢を選択し、そのような他の選択肢をシステムインテグレータに示して付加的な選択を行い、システムインテグレータの選択肢に基づいてソフトウェアを作成する。XMLは、柔軟性および拡張性があるので、使用される。一態様の中でXMLを使用する主な利点は、分散されたリソースの間で結びつくような共通のウェブブラウザを使用して分散環境内でコンフィギュレーションすることができ、そして、システムインテグレータとインタフェースできることである。それによって、インタフェースが特定の構成要素から独立する。

【0021】本発明は好適な実施の形態、ベストモード、および実例として説明され、それらは制限として規定されていない。本発明の更なる対象、特徴、および利点は、添付された図面によって図示されるように、本発明の好適な実施の形態およびベストモードに関する次の詳細な説明からもっと明らかになるだろう。同じ参照番

号は同種の要素を示す。

【0022】

【発明の実施の形態】特定なコンフィギュレーションツールを使ってシステム、方法、ハードウェア、コンピュータ媒体、および、ソフトウェアとして記載されて、既存の作成ツールとソフトウェーデータベースとにインタフェースするシステムインテグレータを手助けし、既存の作成ツールとソフトウェーデータベースとを使用し、また、ウェブブラウザおよびマークアップ言語を使って分散環境までシステムコンフィギュレーションを拡張する。

【0023】以下の記載では、説明目的のため、本発明のより狭い形態に係る具体的な詳細それ自身の利点を理解するだけでなく、本発明のより広い形態の完全な理解を提供するために、多くの具体的な詳細を説明する。しかしながら、本発明のより広い形態が、これらの具体的な詳細を説明せずに、または、本明細書内で明示なく決定される等価物を使って、または、本明細書内で説明されたガイドラインに従って、実行され得ることは、当業者にとって明らかである。公知の技術における本質的でない詳細を使って本発明を必要に曖昧にするのを避けるために、公知の構造および装置がブロック図の形式で示されている。

【0024】本発明の他の形態、特性、および利点は、発明者によって考えられたベストモードを含めて、特別な手段を示す次の詳細な説明から容易に明らかになる。また、本発明は、他のおよび異なる実施の形態も可能であり、その幾つかの詳細は、本発明の趣旨と請求の範囲とから全て逸脱することなく、様々な点で変更なされ得る。図面および説明は実例であり、限定されない。

【0025】図1は、本発明に係る一実施の形態のコンフィギュレーションシステム1におけるハードウェアおよびソフトウェアの概略を示す。コンフィギュレーションシステム1は、コンフィギュレーションツール2、ソフトウェーデータベース3、従来のソフトウェア作成ツール4、公知なウェブサーバ5、システムインテグレータ7とインタフェースするための1つ以上の公知なウェブブラウザ6、および、公知なXSLTプロセッサ8を有している。

【0026】コンフィギュレーションツールは、ここで開示された動作、たとえば、コモン・ゲートウェイ・インタフェース（CGI）またはアクティブ・サーバ・ページ（ASP）の中で開示された動作に従って、スクリプトを使って実行される。ここで開示された動作は、本発明の技術の中で通常技術のうちの一技術の範囲内で十分である。コンフィギュレーションツール2はSCML内でコンフィギュレーションファイルを読み出し、作成ツール4を使用して、（コンフィギュレーションファイル10内で表示されたものから、たとえば以下について）システムインテグレータ7の選択肢に基づいてソフ

トウェアを作成する。コンフィギュレーションツール2またはその一部が、既存のIDE（統合開発環境の頭文字）の一部と統合可能であり、または、既存のIDEの一部であると考えられる。IDEは、1つのインターフェースから一般的に実行される統合ツールであり、たとえば、普通はソフトウェアを開発するためのコンパイラ、エディタ、およびデバッガを含んでいる。

【0027】これらの構成要素2～8のうち全部または1つ以上のサブセットが、1つ以上のネットワークに渡って分散されることもあり、すなわち、図5～図7で示されるように分散環境内に存在することもある。

【0028】XSLは、拡張スタイル/スタイルシート言語の頭文字であり、一種類以上の対象、たとえば表示対象へのプレゼンテーション用に、フォーマッティングを完全なXMLデータに適用できる。XSLは、XMLドキュメント用のフォーマッティングおよびスタイルを具体化するためのXMLボキャブラリーを含んでいる。XSLTはXSL変換、すなわち、他のXMLドキュメント（たとえばHTMLドキュメント）へのXMLドキュメントの変換を示している。XSLTプロセッサは、そのような変換を実行するツールである。

【0029】ソースまたは構成要素ファイル9（完成された実行可能なプログラム）、XML内のコンフィギュレーションファイル10（構成要素の仕様およびパラメータのような情報を有している）、HTMLまたはXMLで書き込まれたウェブページ11（情報を表示するためのテンプレート等として）、および、XSLファイル12がソフトウェアデータベース3内に格納されている。

【0030】コンフィギュレーションファイル10の主要な構成要素はSCMLタグおよびスクリプトを含んでいる。ソフトウェア開発者は、ソフトウェアを開発した時にコンフィギュレーションファイル10を作り出し、コンフィギュレーションファイル10はその時にソフトウェアデータベース3内に格納される。ある場合には、コンフィギュレーションファイルは、本発明に係るソースまたは構成要素ファイルとマイクファイルとから自動的に生成される。

【0031】システムインテグレータと連動することによって、ミドルウェアを作り出すと共に構成要素を統合するためのコンピュータシステムの処理および動作の一実施の形態におけるコンフィギュレーションおよび統合のステップが、他の図面を参照して、図6のフローチャートの中で説明される。

【0032】ステップ600は、統合のタイプに関する複数の選択肢、たとえば、音声コーデック、音響コーデック、ビデオコーデック、音声認識、および、音声合成の選択肢を選択する。これらの選択肢は、スクリプト（インターフェイティビティをカスタマイズし、それをシステムインテグレータに提供するための一組の命令から

構成されるプログラム）によって、ファイルから読み出される。更に詳細には、選択肢を生成する情報が、図1の構成要素ファイル9およびコンフィギュレーションファイル10から、ウェブブラウザ6と、ウェブサーバ5と、コンフィギュレーションツール2のスクリプトとを介して得られる。それから、読み出された選択肢は、ハードウェア仕様の一部として、ウェブページ11およびXSLファイル12の使用によりシステムインテグレータに提供される。好適には、その選択肢は、図7のモニタ701のようなモニタの上に、図2で示されたディスプレイによって提供される。

【0033】ステップ601は、図2のディスプレイから、関連するミドルウェアの選択肢のうちシステムインテグレータ7の選択肢を持ち、そして、何れかのディスプレイがステップ600によって制御される。図2は、システムの統合と所望ミドルウェアの作成との間でアプリケーションタイプと、構成要素と、構成要素のパラメータとを選択する際に、システムインテグレータを手助けするための1つのウェブページにおける一例のディスプレイである。ウェブブラウザ6は、コンピュータシステムとインターフェースするシステムインテグレータとして働き、システムインテグレータの選択が、適切なリンクエリアをクリックするシステムインテグレータによって行なわれる。図2のウェブページでは、リンクエリアは、選択肢に近い陰付き長円体として示されている。

【0034】ステップ602は、ステップ601から入力されたシステムインテグレータに応答し、そのような入力の選択肢がスクリプトを呼び出す。コンフィギュレーションツール2のスクリプトが、今度は、XMLに基づいたコンフィギュレーションファイル10のうち幾つかから、対応するHTMLコードを生成する。XMLに基づいたコンフィギュレーションファイル10は、更に詳細には、構成要素に関しての複数の選択肢と、複数の完成されたソースまたは構成要素の実行可能プログラム、たとえば複数のアプリケーションプログラムの統合用に使用されるそのパラメータの選択肢とを選択および生成するためのSCML記述である。

【0035】ステップ603は、図3で示されるように、構成要素に関しての複数の選択肢とそのパラメータの選択肢とを表示し、そして、何れかのディスプレイがステップ602のHTMLコードで生成される。図3は、更なる選択を行う際にシステムインテグレータを手助けするためのウェブページにおける一例のディスプレイである。更に詳細には、図3によって表示されたウェブページは、構成要素を選択する際に、および、そのパラメータを設定する際にシステムインテグレータを手助けするため、ウェブページ12のフォーマットを使って生成される。3つの予めなされた選択肢が、システムインテグレータにより作成されたように、図3のディスプレイ内で対応するリンクシンボルの各々における中央の

黒色部分によって示されている。

【0036】ステップ604は、システムインテグレータが、ステップ603で表示された選択をし終えるのを待っている。システムインテグレータは、トリガーを使って、たとえば、図3のウェブページ上で「作成」ボタンをクリックすることによって、その選択がなされたことを指示する。

【0037】ステップ605は、ステップ604からの入力に応答して、幾らか多くのスクリプトを呼び出し、呼び出されたスクリプトを実行してミドルウェアを作成する。コンフィギュレーションツール2のスクリプトは、今度は、XMLに基づかれたコンフィギュレーションファイル10の幾らかから、対応するHTMLコードを生成する。XMLに基づいたコンフィギュレーションファイル10は、更に詳細には、統合タイプと、構成要素と、複数の完成された実行可能なプログラムの統合のために利用される構成要素のパラメータとの選択肢に対応するミドルウェアの仕様を選択および生成するためのSCML記述である。更に詳細には、ミドルウェアの仕様を生成する情報は、図1で示されるように、記憶装置からウェブブラウザ6と、ウェブサーバ5と、コンフィギュレーションツール2のスクリプトとを介して得られる。

【0038】ステップ606は、ステップ605内で生成されたミドルウェアの仕様をシステムインテグレータ7に提供する。好適には、ミドルウェアの仕様は、図7のモニタ701上に、図4に示されたウェブページ11を表示することにより提供される。

【0039】ステップ607は、システムインテグレータ7が、ステップ606で表示されたミドルウェアの仕様を承認するのを待っている。承認は、システムインテ

グレータ7によって、図7のモニタ701上で示されるよう、図4のディスプレイ内の「DOWNLOAD NOW (今すぐダウンロード) …」ボタンをクリックすることでなされる。

【0040】ステップ608は、ステップ607から入力されたシステムインテグレータに応答して、コンフィギュレーションツールのスクリプトを呼び出す。構成要素が前のステップで、たとえばステップ605および／またはステップ602で既にダウンロードされていなかった場合には、コンフィギュレーションツールは、その時に、統合されるべき完成された実行可能なプログラムをダウンロードする。ダウンロードされた構成要素は、システムインテグレータによって行われた選択によって決定、すなわち、その選択に左右される。スクリプトの実行は既存の作成ツール4を利用してミドルウェアを作成する。そのミドルウェアは、ミドルウェアのタイプを決定する選択肢、構成要素を選択する選択肢、および、パラメータを設定する選択肢を含めて、そして、システムインテグレータによってどの選択がなされるのかを含めて、その選択肢に基づいて完成された実行可能なプログラムを統合する。それによって、システムインテグレータは、ステップ601、604、607から受け取られた選択肢に依存して、統合されたアプリケーションプログラムまたはミドルウェアを作成する際に、コンピュータシステムによって手助けされる。

【0041】次の実例1は、図1の一例のコンフィギュレーションファイル10における冒頭部分から取られ、いま問題になっている実例では、図6のステップ600に従って、図2のディスプレイを生成する。

【0042】

【実例1】

【実例1】

**EXAMPLE 1**

```
:
:
<MIDDLEWARE>
  <CATEGORY>Speech Codecs</CATEGORY>
  <NAME>ITU-T G.XXX</NAME>
  <VERSION> 1.00</VERSION>
  <RELEASEDATE>
    <YEAR>200 1</YEAR><MONTH>5</MONTH><DAY> 11</DAY>
  </RELEASEDATE>
  <AUTHOR>XYZ, Ltd. R&D Div.</AUTHOR>
  <COPYRIGHT> XYZ, Ltd.</COPYRIGHT>
:
:
```

【0043】次の実例2は、図1の一例のコンフィギュレーションファイル10であり、いま問題となっている実例では、図6のステップ602、603に従って、図

3のディスプレイを生成する。

【0044】

【実例2】

## 【実例2】

## EXAMPLE 2

```

<CHOICES type="PROCESSOR" option="cpu">
<ITEM option_value="sh3">CPU3</ITEM>
<ITEM option_value="sh3dsp">CPU3D SP</ITEM>
<ITEM option_value="sh4">CPU4</ITEM>
<ITEM option_value="sh5c">CPU5 COMPACT MODE</ITEM>
<ITEM option_value="sh5m">CPU5 MEDIA MODE</ITEM>
</CHOICES>

<CHOICES type="ENDIANNESS" option="endian">
<ITEM option_value="big">BIG ENDIAN</ITEM>
<ITEM option_value="little">LITTLE ENDIAN</ITEM>
</CHOICES>

<CHOICES type="OPTIMIZATION">
<ITEM option="speed">HIGHER SPEED, BIGGER CODE SIZE</ITEM>
<ITEM option="nospeed">MEDIUM SPEED & CODE SIZE</ITEM>
<ITEM option="size">LOWER SPEED, SMALLER CODE SIZE</ITEM>
</CHOICES>

<INTERFACE type="FILE">gXXXapi.h</INTERFACE>

<COMPONENT_LIST>
  <COMPONENT
    type="NAME">XXX_TOP_COMPONENT</COMPONENT>
    <COMPONENT type="NAME">COMPONENT_AA</COMPONENT>
    <COMPONENT type="NAME">COMPONENT_BB</COMPONENT>
    <COMPONENT type="NAME">COMPONENT_CC</COMPONENT>
    <COMPONENT>
      <NAME>COMPONENT_DD</NAME>
      <CHOICES>
        .....
      </CHOICES>
    </COMPONENT>
  </COMPONENT_LIST>

```

【0045】上述の実例1および実例2のコンフィギュレーションファイル10は、完全なミドルウェアのプログラムの一部を具体化するためのものである。MIDDLEWAREタグは最上層を表示し、CATEGORYタグ、NAMEタグ、VERSIONタグ、RELEASEDATEタグ、AUTHORタグ、COPYRIGHTタグ、任意個のCHOICESタグ、INTERFACEタグ、および、COMPONENT LISTタグを含めて様々なタグを含んでいる。三種類のCHOICES、すなわちプロセッサ、エンディア

ン、および最適化（最適化のレベル）が示され、いま問題になっている実例では選択肢の数はそれぞれ5つ、2つ、および3つ（選択肢としてNONEを数えない）である。

【0046】エンディアンとは、1バイトより大きな値の複数バイトがメモリ内に格納される順番を意味する。たとえば、エンディアンは整数値およびポインターに影響を及ぼし、そして一方、1バイト文字セットのアレイ

は影響を及ぼされない。エンディアンは、ハードウェア、特に、プロセッサの選択(CPU)に依存する。最も一般的な2つのタイプのエンディアンが存在する:

1) リトルエンディアンマシンは、最小のメモリアドレス上に最小桁のバイトを格納するので、ワードがリトルエンドファーストで(小さい方から順に)格納される; 2) ビッグエンディアンマシンは、最小のメモリアドレス上に最大桁のバイトを格納するので、ワードがビッグエンドファーストで(大きい方から順に)格納される。プロセッサの中にはビッグエンディアンまたはリトルエンディアンのモードで動くことができるものもある。マシンの中にはリトルエンディアンマシンまたはビッグエンディアンマシンとして選択的に動作するものもある。

【0047】最適化は、データ構造の選択および設計、アルゴリズム、および、命令シーケンスを介して、図2および図3のディスプレイからステップ601およびステップ604の中で選択された構成要素と、コードサイズと、ミドルウェアタイプの実行速度とに関して、最も効率的に統合するプログラムのためにステップ605で実行されるプロセッサである。最適化の選択は、効率的で実行可能なコードを作り出す際に、システムコンパイラまたはアセンブラーの処理を決定する。たとえば、最適化は、図3のディスプレイ(ステップ604)から選択された最適化に応答して、ステップ604内で選択されたプロセッサ上で効率的に動くステップ605またはステップ608に従ってダウンロードされた完成された実

行可能なプログラムを準備する最適化コンパイラを使って得られる。

【0048】タイプの属性を使って、(図2からの)ミドルウェアと、プロセッサと、エンディアンと、最適化と、コード内で例示されないRTOS(リアルタイムオペレーティングシステム)等とのタイプを示す。CHOICES内のITEMタグは、選択肢として、たとえば実例では図3としてウェブブラウザ上に表示されたラベルを表現する。オプションの属性とオプション\_値の属性とは、コンパイルまたはアセンブリするためのオプションを設定するのに使用される。INTERFACEタグは、ミドルウェアのAPI(アプリケーション・プログラミング・インターフェース)を説明する幾つかのドキュメントを含み、オプションである。そのドキュメントは、単にあるインターフェース機能と他の一般的な情報を定義するヘッダーファイルであることがある。COMPONENT LISTは、構成要素名またはXML構造の何れかによって表現され得る幾つかの構成要素から成る。

【0049】以下、実例3では、構成要素用に、図1の一例のコンフィギュレーションファイル10が示されている。このコンフィギュレーションファイルは、ステップ605を作成するミドルウェアの一部として、図6のコンフィギュレーションツール2のスクリプトによって処理される。

【0050】

【実例3】

## 【実例 3】

```

EXAMPLE 3
:
:
<COMPONENT>
  <NAME>COMPONENT_AA</NAME>
  <INTERFACE type="file">componentAA.h</INTERFACE>
  <CHOICES>
    <IMPLEMENT type="SOFT_ONLY">
      <ITEM>CPU3</ITEM>
      <SOURCE type="C_FILE">
        define="CPU3" componentAA.c</SOURCE>
      </IMPLEMENT>
      <IMPLEMENT>
        <ITEM>CPU3D 8P</ITEM>
        <SOURCE type="C_FILE">
          define="CPU3D8P" componentAA.c</SOURCE>
        <SOURCE type="ASM_FILE">componentAAh3dep.asm</SOURCE>
      </IMPLEMENT>
      <IMPLEMENT>
        <ITEM>CPU4</ITEM>
        <SOURCE type="C_FILE">
          define="CPU4" componentAA.c</SOURCE>
        <SOURCE type="ASM_FILE">componentAAh4.asm</SOURCE>
      </IMPLEMENT>
      <IMPLEMENT>
        <ITEM>CPU5 COMPACT MODE</ITEM>
        <SOURCE type="C_FILE">
          define="CPU5COMPACT" componentAA.c</SOURCE>
        </IMPLEMENT>
        <IMPLEMENT>
          <ITEM>CPU5 MEDIA MODE</ITEM>
          <SOURCE type="C_FILE">
            define="CPU5MEDIA" componentAA.c</SOURCE>
          </IMPLEMENT>
          <IMPLEMENT type="HARD_DRIVER">
            <ITEM>ANY_PROCESSOR</ITEM>
            <SOURCE type="C_FILE">componentAA.c</SOURCE>
          </IMPLEMENT>
          <IMPLEMENT type="ASIC">
            <SOURCE type="C_FILE">componentAAasic.c</SOURCE>
          </IMPLEMENT>
        </CHOICES>
      </COMPONENT>
:
:

```

【0051】 上述の実例3のコンフィギュレーションファイル10では、全プログラムの一部である個々の構成要素に対して、COMPONENTタグは、全プログラムと全く同様に、最上層を表現し、NAMEタグ、INTERFACEタグ、およびCHOICESタグを含む。表示「AA」は、指定されたプロセッサである。幾つかのIMPLEMENTタグは、CHOICESタグ内に含まれている。SOURCEタグは、「タイプ」属性をソースまたは構成要素タイプ（必要な場合には、このコンフィギュレーションファイルの一部としてソースまたは構成要素コード自身を含むことができるもの）に

表示した状態で、ソースまたは構成要素ファイルの名前を示す。「定義」属性は、コンパイルプリプロセッサによって必要とされる従来の#defineコマンド（または、-define =valueまたは-Dvalueのようなコンパイルингオプション）の中の値に対応する。

【0052】 以下、実例4では、幾つかのプロセッサ用に分割された一例のCソースコードファイル11を示す。このソースファイルは、図6のステップ605を作成するミドルウェアの一部として、作成ツールの1つであるコンパイラによりコンフィギュレーションツール2

を介して処理される。

【0053】

【実例4】

```
EXAMPLE 4

;

#if defined      CPU3 DSP
int function() /* optimized for CPU3DSP*/
{
;

;

}

#elif defined    CPU4
int function() /* optimized for CPU4*/
{
;

;

}

#elif defined    CPU5COMPACT
int function() /* optimized for CPU5 compact mode */
{
;

;

}

#elif defined    CPU5MEDIA
int function() /* optimized for CPU5 media mode */
{
;

;

}

#else
int function() /* not optimized for any specific processor */
{
;

;

}

#endif
```

【0054】Cソースまたは構成要素コードにおける上述の実例4では、「コンポーネントAA.c」は幾つかのプロセッサ用に共通して使用される。この構成要素のコンフィギュレーションファイル (COMPONENT) のコンテンツは、上述の実例2のコンフィギュレーションファイル10の中で、COMPONENT\_DDのようなMIDDLEWARE定義ファイル内に含まれていることがある。より簡単なメンテナンスのために、構成要素用の分離ファイルを作ること

は更に有益である。また、プログラムソースまたは構成要素コードを分離ファイル内に維持することが推奨される。

【0055】XMLまたは類似言語の拡張特性は、統合するためのシステムの一部としてネットワークに接続された環境、または、分散された環境内で構成するのに特に有用である。そのような分散環境は図5および図7で示され、そこでは要素が図1に関して既に記述されてい

る。本実施の形態では、図5のソフトウェアデータベースサーバは従来から利用されている。ネットワークに接続することは、図面で示されたカップリング（結合すること）から明らかであり、そこでは次の構成要素：システムインテグレータ7、クライアント、コンフィギュレーションサーバ、およびソフトウェアデータベースサーバが局所的であることもあり、または、LANまたはWANのようなネットワークによって結合された遠方場所であることもある。すなわち、構成要素は、それぞれ、他の個々の構成要素の物理的な場所から独立していることもある。

【0056】従って、本発明の主要な利点は、図5および図7のような分散環境内でミドルウェアをコンフィギュレーションできる能力であり、その能力は、コンフィギュレーションサーバマシンを使って、ネットワークに接続された環境内に全て分散された1つ以上のクライアントマシンと1つ以上のソフトウェアデータベースサーバマシンとをシステムインテグレータがどのようにコンフィギュレーションするのかを示している。たとえば、ソフトウェア開発セクションのそれぞれ（たとえば、図5の実施の形態でのコンフィギュレーションサーバ）が、分離されたソフトウェアデータベースサーバマシンを管理して、対応するソフトウェアをタイムリーに体系化された方法で維持および改訂する。

【0057】本発明を利用できる様々な方法が存在する。それは、販売、請求書の作成、および、警備のような役目に関係される会社内の販売員（たとえば販売および管理人）と取引先とにミドルウェアを提供できる。本発明を直接の利用により、大規模で世界的な会社における地形上の様々な場所で、異なるグループに渡ってミドルウェア開発活動の広がりが合理化される。本発明は、異なるグループによって正当と認められると共に予め検証を行われた構成要素、すなわち、完成された実行可能なプログラムの再利用を重要視することにより、構成要素として統合されたプログラムを作成するようにミドルウェア開発を変えることができ、それ故に、構成要素の細部ではなく、システムの問題にシステムインテグレータを集中させることができる。

【0058】実施の形態に係る統合方法を実施するためのコードを保持するコンピュータ読み出し可能媒体は、本発明に係るコードたとえばスクリプトを実行用プロセッサに提供することに関与する何らかの媒体である。実例は揮発性または不揮発性の媒体を含む。不揮発性の媒体は、たとえば、光学的ディスクまたは磁気的なフレキシブルディスクまたはテープ、およびハードディスク、更に詳細には、CD-ROM、CD RW、およびDVD：パンチカード、ペーパーテープ、光学的マークシート、または、穴パターンを有する何らかの物理的媒体を含む。一般に、ここで利用されているようにコンピュータ読み出し可能媒体は、一時的であろうともっと永続的

であろうと、何らかの物理的な固着を含み、その物理的な固着からコンピュータはコードを読み出すことができる。

【0059】実施の形態における伝達ラインは、同軸ケーブル、銅ワイヤ、ワイヤレスリンク、および、光ファイバを含み、それらは、無線周波数（RF）および赤外線（IR）通信の間に生成されるような音波、オプティカルウェーブ（光学波）、または電磁波を送ることもある。

【0060】本発明は、多くの実施の形態、手段、変更、および、それらに特有な利点を有する変化に関して記述されたが、必ずしもそのように限定されず、より広い形態に従った様々に明白な変更と等価な配置とに及び、そして、請求項の趣旨および範囲に従う全てに及んでいる。

【0061】

【発明の効果】本発明の主要な利点は、分散環境内でミドルウェアをコンフィギュレーションできる能力であり、その能力は、コンフィギュレーションサーバマシンを使って、ネットワークに接続された環境内に全て分配された1つ以上のクライアントマシンと1つ以上のソフトウェアデータベースサーバマシンとをシステムインテグレータがどのようにコンフィギュレーションするのかを示している。たとえば、ソフトウェア開発セクションのそれぞれが、分離されたソフトウェアデータベースサーバマシンを管理して、対応するソフトウェアをタイムリーに体系化された方法で維持および改訂する。

【図面の簡単な説明】

【図1】本発明に係るハードウェアおよびソフトウェアコンフィギュレーションシステムを示す図である。

【図2】システムの統合と所望ミドルウェアの作成との間に構成要素を選択する際に、システムインテグレータを手助けするウェブページの一例の表示を示す図である。

【図3】統合用の構成要素のパラメータを設定するため更に選択する際に、システムインテグレータを手助けするウェブページの一例の表示を示す図である。

【図4】システムインテグレータの選択肢に基づいてコンフィギュレーションされたミドルウェアのうち、システムインテグレータが確認すれば直ちにダウンロードおよび作成される一例の表示を示す図である。

【図5】分散された環境内への本発明に係る実施の形態の拡張として、分散されたハードウェアおよびソフトウェアのコンフィギュレーションシステムを示す図である。

【図6】システムインテグレータとインタフェースすることにより、ミドルウェアを作り出すため、かつ、構成要素を統合するためのコンピュータシステムの処理および動作における一実施の形態のフローチャートである。

【図7】分散環境内への本発明に係る実施の形態の拡張

として、他の分散されたハードウェアおよびソフトウェアのコンフィギュレーションシステムを示す図である。

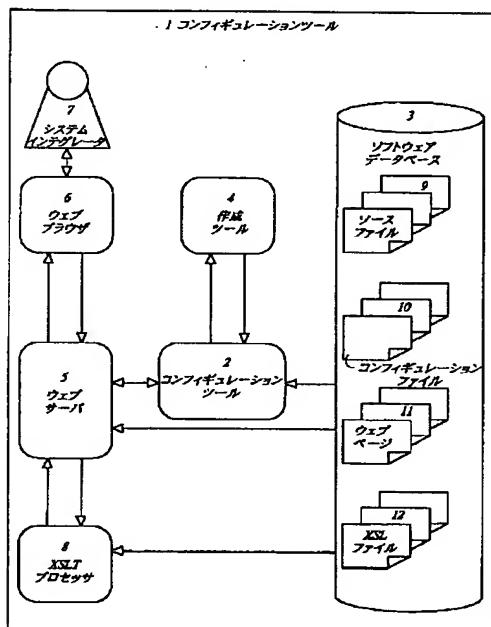
【符号の説明】

- 1 コンフィギュレーションシステム
- 2 コンフィギュレーションツール
- 3 ソフトウェアデータベース
- 4 作成ツール
- 5 ウェブサーバ

- 6 ウェブブラウザ
- 7 システムインテグレータ
- 8 XSLT プロセッサ
- 9 ソースファイル
- 10 コンフィギュレーションファイル
- 11 ウェブページ
- 12 XSL ファイル

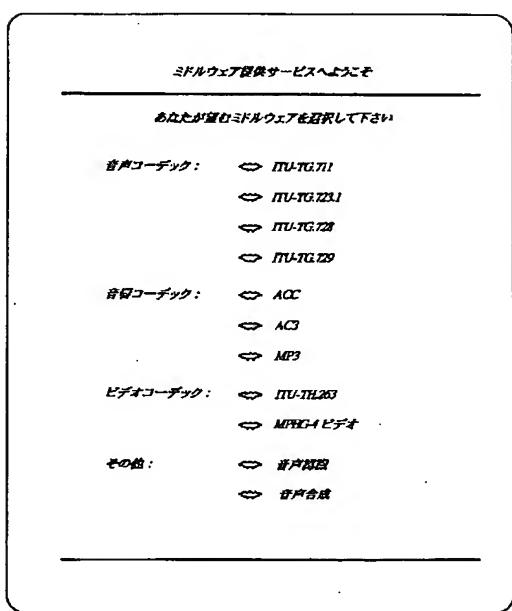
【図 1】

図 1



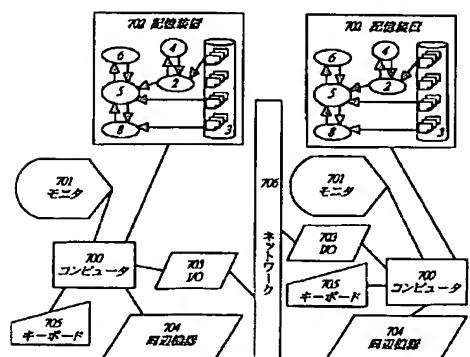
【図 2】

図 2



【図 7】

図 7



【図3】

図3

ミドルウェアの音声コーデックITU-TGXXX用のパラメータを選択してください

プロセッサ:  SH3  
 SH3 DSP  
 SH4  
 SH5コンパクトモード  
 SH5メディアモード

エンディアン:  ビッグ  
 リトル

最適化:  高速度、コードサイズ大  
 中間速度とコードサイズ  
 低速度、コードサイズ小  
 なし

利用可能なハードウェア:  コンボーネント (オプション)

【図4】

図4

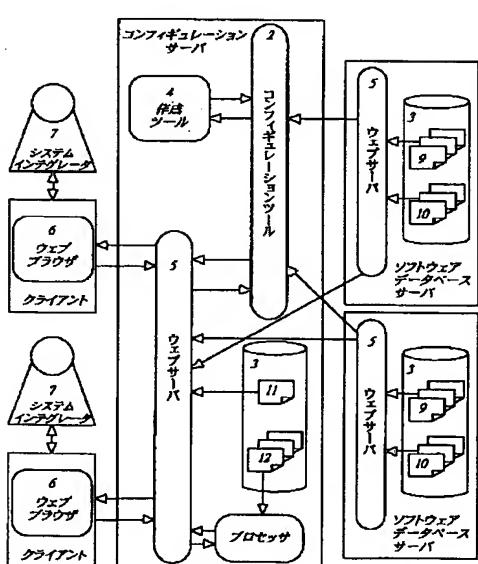
あなたの数値化したミドルウェアの仕様は:

名前: ITU-TGXXX  
バージョン: 1.00  
プロセッサ: SH5メディアモード  
エンディアン: ビッグ  
最適化: 中間速度とコードサイズ

ミドルウェアのオブジェクトファイルとそのAPIを示すヘッダーファイル、各段のドキュメント(ある場合)とを含んだ正方形ファイルは現在ダウンロード可能です。

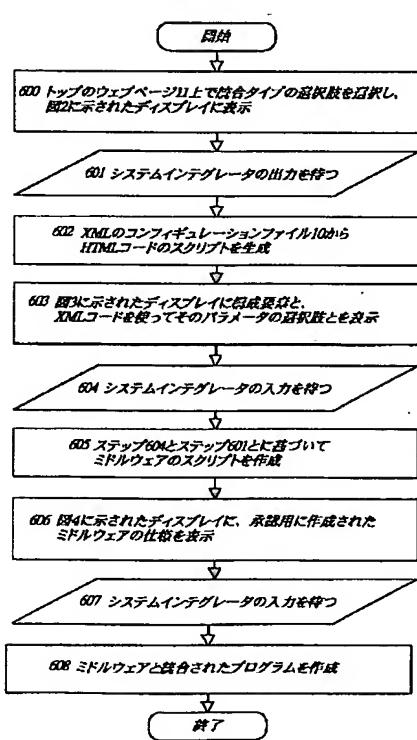
【図5】

図5



【図6】

図6



フロントページの続き

F ターム(参考) 5B076 AB10 DB01 DC02 DD06 DD10  
DF06 EC07